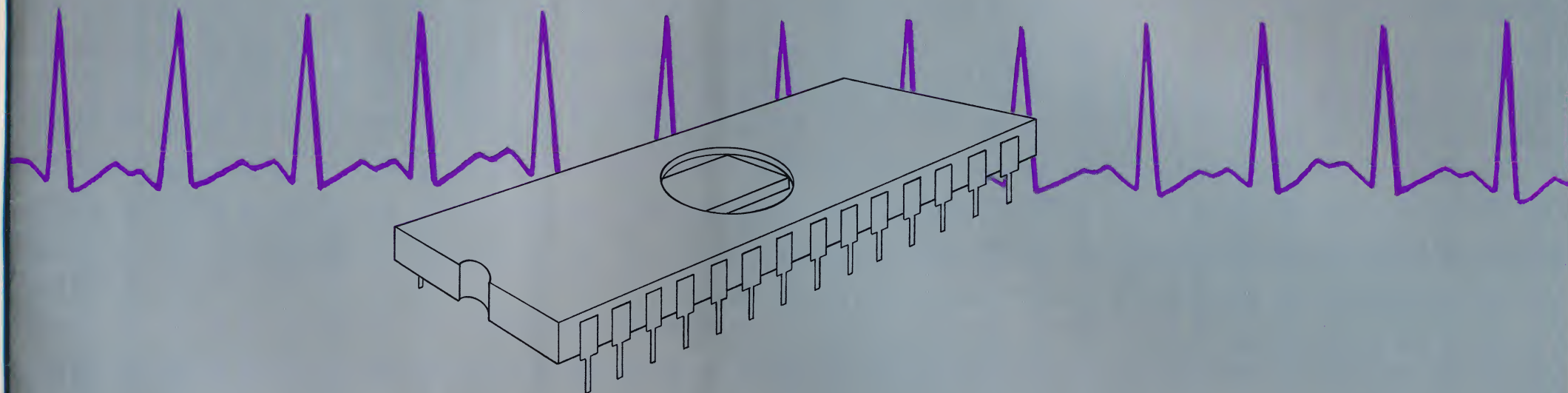
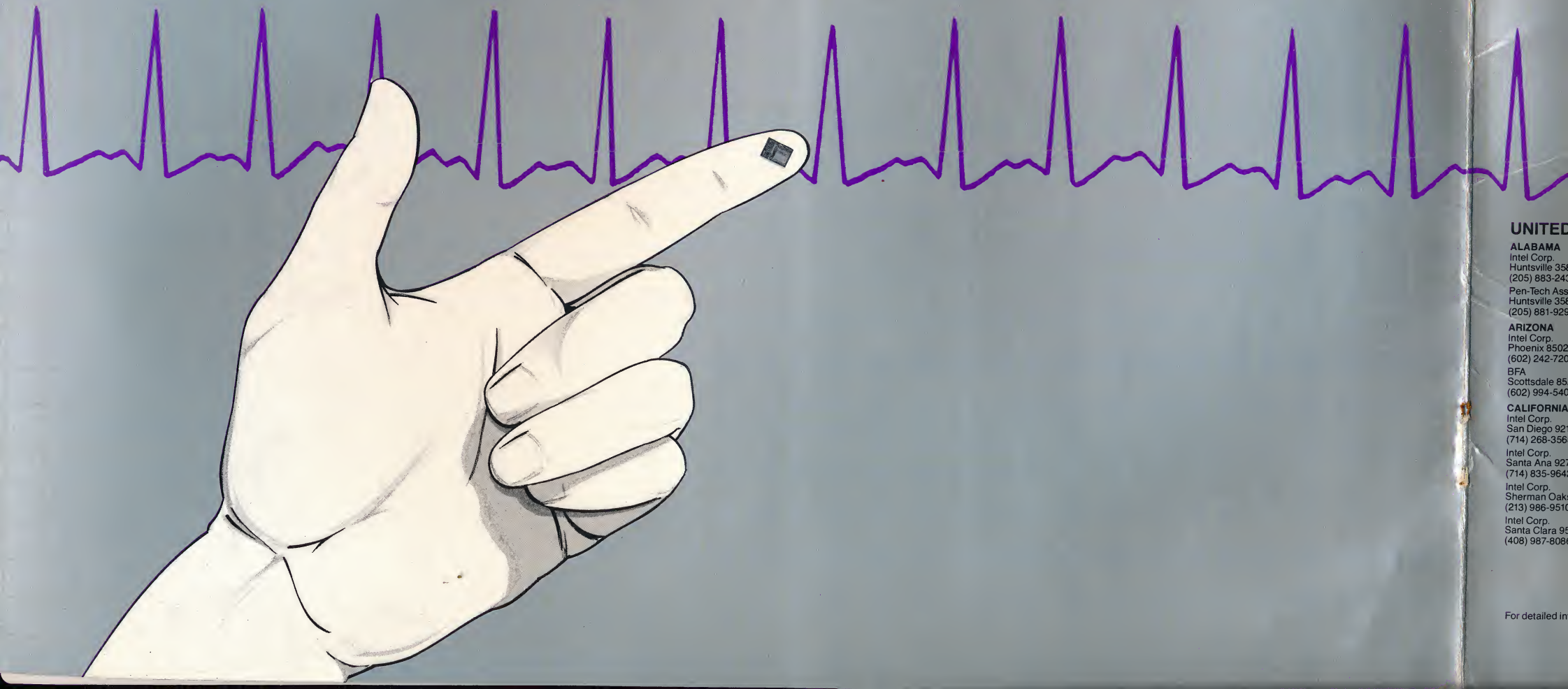


JUNE 80

2920 Signal Processor: Breakthrough in Analog Design



intel® delivers.



UNITED

ALABAMA
Intel Corp.
Huntsville 358
(205) 883-243
Pen-Tech Ass
Huntsville 358
(205) 881-929

ARIZONA
Intel Corp.
Phoenix 8502
(602) 242-720
BFA
Scottsdale 85
(602) 994-540

CALIFORNIA
Intel Corp.
San Diego 921
(714) 268-356
Intel Corp.
Santa Ana 927
(714) 835-964
Intel Corp.
Sherman Oaks
(213) 986-9510
Intel Corp.
Santa Clara 95
(408) 987-808

For detailed in

UNITED STATES AND CANADIAN SALES OFFICES

ALABAMA

Intel Corp.
Huntsville 35802
(205) 883-2430
Pen-Tech Associates, Inc.
Huntsville 35801
(205) 881-9298

ARIZONA

Intel Corp.
Phoenix 85021
(602) 242-7205
BFA
Scottsdale 85251
(602) 994-5400

CALIFORNIA

Intel Corp.
San Diego 92111
(714) 268-3563
Intel Corp.
Santa Ana 92701
(714) 835-9642
Intel Corp.
Sherman Oaks 91403
(213) 986-9510
Intel Corp.
Santa Clara 95051
(408) 987-8086

CALIFORNIA (continued)

Earle Associates, Inc.
San Diego 92111
(714) 278-5441
Mac-I
Berkeley 94704
(415) 843-7625
Mac-I
Cupertino 95014
(408) 257-9880
Mac-I
Fountain Valley 92708
(714) 839-3341
Mac-I
San Francisco 94104
(415) 982-3673
Mac-I
Woodland Hills 91364
(213) 347-5900
COLORADO
Intel Corp.
Denver 80222
(303) 758-8086
Westek Data Products, Inc.
Evergreen 80439
(303) 674-5255
Westek Data Products, Inc.
Boulder 80302
(303) 449-2620
Westek Data Products, Inc.
Littleton 80210
(303) 797-0482

CONNECTICUT

Intel Corp.
Danbury 06810
(203) 792-8366

FLORIDA

Intel Corp.
Ft. Lauderdale 33309
(305) 771-0600

Intel Corp.
Orlando 32804
(305) 628-2393

Pen-Tech Associates, Inc.
Deerfield Beach 33441
(305) 421-4989

Pen-Tech Associates, Inc.
Maitland 32751
(305) 645-3444

GEORGIA

Pen-Tech Associates, Inc.
Marietta 30060
(404) 424-1931

ILLINOIS

Intel Corp.
Rolling Meadows 60008
(312) 981-7200
First Rep Company
Chicago 60656
(312) 992-0830

ILLINOIS (continued)

Technical Representatives, Inc.
Bloomington 61701
(309) 829-8080

IOWA

Technical Representatives, Inc.
Cedar Rapids 52405
(319) 393-5510

KANSAS

Intel Corp.
Overland Park 66210
(913) 642-8080

Technical Representatives, Inc.
Lenexa 66214
(913) 888-0212, 3 & 4

KENTUCKY

Lowry & Associates, Inc.
Lexington 40555
(606) 273-3771

MARYLAND

Intel Corp.
Hanover 21076
(301) 796-7500
Glen White Associates
Timonium 21093
(401) 252-6360
Mesa, Inc.
Rockville 20852
(301) 881-8430 (Wash.)
(301) 792-0021 (Balt.)

MASSACHUSETTS

Intel Corp.
Chelmsford 01824
(617) 667-8126
EMC Corp.
Newton 02164
(617) 244-4740

MICHIGAN

Intel Corp.
Southfield 48075
(313) 353-0920

Lowry & Associates, Inc.
Brighton 48116
(313) 227-7067

Lowry & Associates, Inc.
Grand Rapids 49505
(616) 363-9839

MINNESOTA

Intel Corp.
Edina 55435
(612) 835-6722
Dytek North
St. Paul 55104
(612) 645-5816

MISSOURI

Technical Representatives, Inc.
Hazelwood 63042
(314) 731-5200

NEW JERSEY

Intel Corp.
Edison 08817
(201) 494-5040

NEW MEXICO

BFA Corporation
Las Cruces 88001
(505) 523-0601
BFA Corporation
Albuquerque 87111
(505) 292-1212

NEW YORK

Intel Corp.
Hauppauge 11787
(516) 231-3300
Intel Corp.
Poughkeepsie 12601
(914) 473-2303
Intel Corp.
Rochester 14606
(716) 328-7340
Measurement Technology, Inc.
Great Neck 11021
(516) 482-3500
T-Squared
Syracuse 13206
(315) 463-8592
T-Squared
Victor 14564
(716) 924-9101

NORTH CAROLINA

Pen-Tech Associates, Inc.
Highpoint 27260
(919) 883-9125
Glen White Associates
Raleigh 27609
(919) 787-7016

OHIO

Intel Corp.
Dayton 45415
(513) 890-5350

Intel Corp.
Cleveland 44122
(216) 464-2736

Lowry & Associates, Inc.
Cleveland 44134
(216) 398-0506

Lowry & Associates, Inc.
Dayton 45432
(513) 429-9040

Lowry & Associates, Inc.
Columbus 43229
(614) 436-2051

OREGON

Intel Corp.
Beaverton 97005
(503) 641-8086

PENNSYLVANIA

Intel Corp.
Fort Washington 19034
(215) 542-9444
Lowry & Associates, Inc.
Pittsburgh 15520
(412) 922-5110
Q.E.D. Electronics
Hartboro 19040
(215) 674-9600

TEXAS

Intel Corp.
Dallas 75234
(214) 241-9521
Intel Corp.
Houston 77087
(713) 784-3400

Industrial Digital Systems Corp.
Houston 77036
(713) 988-9421

Intel Corp.
Austin 78752
(512) 454-3628

VIRGINIA

Glen White Associates
Charlottesville 22901
(804) 295-7686
Glen White Associates
Lynchburg 24506
(804) 384-6920

VIRGINIA (continued)

Glen White Associates
Colonial Beach 22442
(804) 224-7764

WASHINGTON

Intel Corp.
Bellevue 98005
(206) 453-8086

WISCONSIN

Intel Corp.
Milwaukee 53207
(414) 747-0789
First Rep Company
Milwaukee 53227
(414) 546-2033

CANADA

Intel Semiconductor Corp.
Ottawa, Ontario K2H 8R2
(613) 829-9714
Intel Semiconductor Corp.
Mississauga, Ontario L4V 1E3
(416) 671-0611
Multitek, Inc.
Ottawa, Ontario K2G 0G3
(613) 226-2365
Multitek, Inc.
Toronto
(416) 245-6422
Multitek, Inc.
Montreal
(514) 481-1350

For detailed information on any of the 2920 family of products, contact any Intel franchised distributor: Arrow Electronics, Alliance, Almac/Stroum, Avnet Electronics, Component Specialties, Hamilton/Avnet, Harvey, Industrial Components, L.A. Varah, Pioneer, Wyle, Zentronics.

- ☐ Please tell me more about Intel's LSI Breakthrough for Analog Design. Have an Intel Sales Engineer phone me as soon as possible. My number is () _____
- ☐ Please send me additional information about Intel's 2920 Signal Processor.

Name _____

Title _____

Company _____

Address _____ Mail Stop _____

City/State/Zip _____

Country _____

- ☐ Please tell me more about Intel's LSI Breakthrough for Analog Design. Have an Intel Sales Engineer phone me as soon as possible. My number is () _____
- ☐ Please send me additional information about Intel's 2920 Signal Processor.

Name _____

Title _____

Company _____

Address _____ Mail Stop _____

City/State/Zip _____

Country _____



NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY CARD

FIRST CLASS PERMIT NO. 621 SANTA CLARA, CA

POSTAGE WILL BE PAID BY ADDRESSEE

Intel Corporation
3065 Bowers Avenue
Santa Clara, CA 95051

ATTENTION: Literature Department



NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY CARD

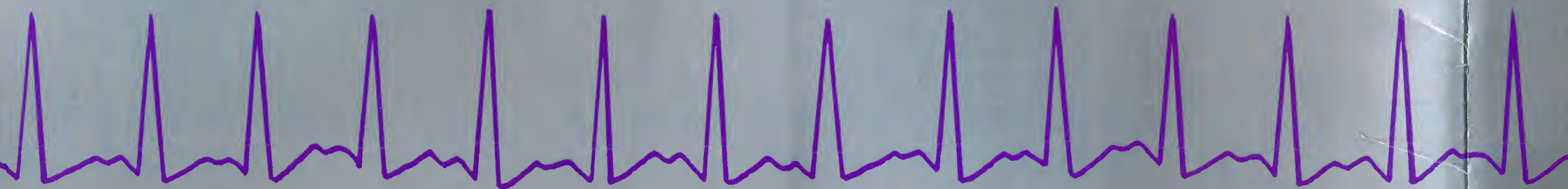
FIRST CLASS PERMIT NO. 621 SANTA CLARA, CA

POSTAGE WILL BE PAID BY ADDRESSEE

Intel Corporation
3065 Bowers Avenue
Santa Clara, CA 95051

ATTENTION: Literature Department





Intel, Intellec, INSITE, Library Manager, MCS,
MEGACHASSIS, MICROMAP PROMPT, ISBC,
MULTIBUS, RMX-80, UPI-41, and ICE are trademarks
of Intel Corporation.

Intel Corporation assumes no responsibility for the
use of any circuitry or software other than circuitry
or software embodied in an Intel product. No other
circuit patent licences are implied.

Intel Corporation 1980

Printed in USA/K64/01-1-80/50K/GR#S

LSI Breakthrough for Analog Design

Large Scale Integration of electronic circuits is revolutionizing hundreds of industries, thousands of applications. For most companies, LSI meant the opportunity to change their competitive position, bringing to market dramatic improvements in product size, cost and manufacturing efficiency.

But the LSI revolution has also changed the rules for designers. LSI brought the pressure to develop systems with more features, at a faster pace, with lower cost and greater reliability.

To help streamline the designer's task, Intel introduced the first microcomputer in 1971. This intelligent LSI component made practical the concept of programmability and general purpose microelectronics. The microprocessor also made it possible for designers to use one component for a wide variety of digital applications,

allowing the same set of building blocks to be arranged in an almost endless number of configurations. The effect has been to reduce cost and make LSI economical even for small volume applications. More importantly, the microcomputer freed the designer to create innovative solutions more rapidly and more efficiently.

Introducing the 2920 Signal Processor

Today, Intel brings the advantages of LSI to the analog world: our 2920 Signal Processor, the first general purpose analog system on a chip. Plus a complete computer-aided development package to help speed your systems to market faster than ever before.

The 2920 approach makes possible a true revolution in analog system design, bringing the power of programmability to analog designers.

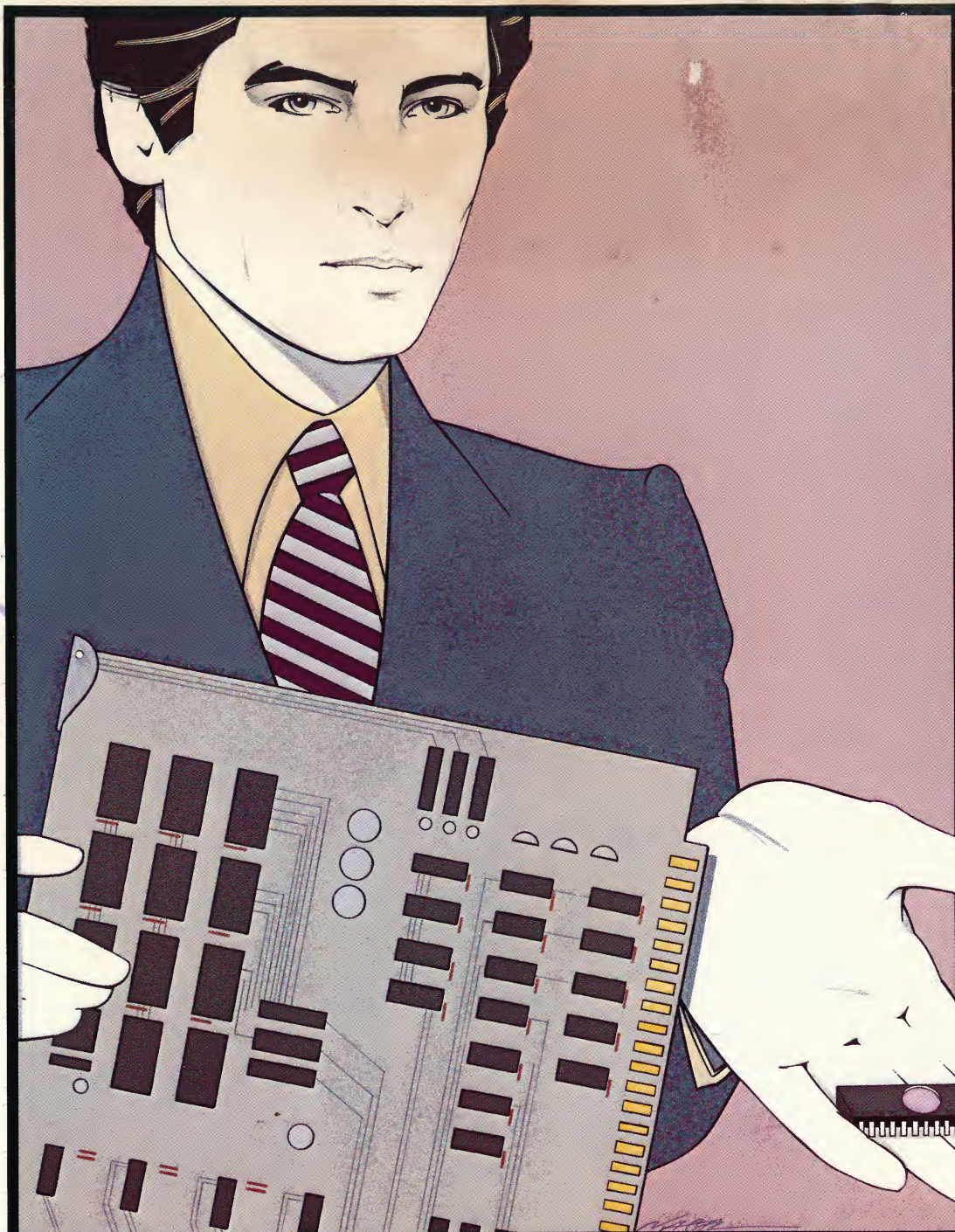
Packing the equivalent of some 18,000 transistors on a single quarter-inch chip and operating hundreds of times faster than digital microcomputers, the 2920 allows you to implement real-time signal processing systems using a self contained, programmable processor.

The single-chip 2920 eliminates the need for costly components like precision resistors and capacitors. Circuit

performance remains consistent from one production lot to another. And performance degradation over time due to circuit changes or noise is virtually eliminated.

For the first time, you can develop and implement analog designs using Intel's development system and support software. And the final design solution can literally be placed at your fingertips.

The advantages are obvious. Because of its size, the 2920 can fit into spaces too compact for analog systems on a board. Because it is programmable, you can use Intel's Development System hardware and software to significantly speed your product development and time-to-market. Finally, because the 2920 is a solid-state device produced with Intel's proven NMOS process, you're assured of reliability and manufacturing repeatability to a degree not possible with traditional analog designs.



Intel's 2920: A revolution in analog design

Because it is programmable, the 2920 Signal Processor can be used as a total-system-solution for literally thousands of different analog applications. The 2920 also opens up new applications, reducing what would be a board full of components to a single device.

In short, the single chip 2920 delivers the broad capability to implement single complex systems, such as modems, or several less complex circuits, such as filters and detectors. Other functions easily programmable with the 2920 include: limiters, rectifiers, multipliers, dividers, automatic gain controls and more.

How the 2920 works

Our 2920 is designed to operate as a single-chip, self-contained system. It embodies an architecture and instruction set developed especially for precision analog signal processing. It can perform all functions needed by a sampled data system. An on-board Erasable, Programmable Read

Only Memory (EPROM) stores the program which controls all 2920 functions. This EPROM will store up to 192 instructions, each 24 bits long. These 24 bits are split into several fields, with each field controlling a subsystem of

the 2920. In order to maintain a constant sample rate, the program is executed sequentially, with no conditional branches.

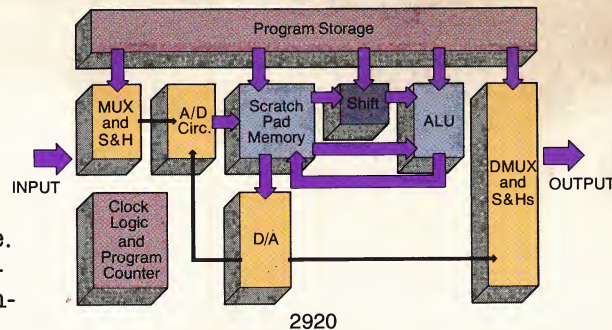
The sample rate is determined by the length of the program and the instruction cycle time—400 nsecs at the maximum 10-MHz clock rate. A full 192 instruction program, running at a 10-MHz

clock rate will yield a 13-kHz sample rate. This allows processing of a signal bandwidth of approximately 4 kHz (maximum of 6.5 kHz with a rectangular filter). Shorter programs will have proportionately higher sample rates.

Device operations

Although many configurations are possible, operation of the 2920 can best be demonstrated by following an analog input signal through the Signal Processor subsystems until it appears at the output as a processed analog signal. Under program control, one of the four possible inputs is selected and the signal sampled and held. This signal is then converted to a digital word with up to 9 bits of linear conversion (sign bit and 8 amplitude bits). The bits formed by the successive approximation A/D conversion are stored in the Digital-Analog Register

(DAR) until the conversion is complete. This register is the interface between the analog and the digital sections of



subsystem shown in the block diagram includes the 2-port addressable RAM with forty 25-bit words, a binary shifter, and the 28-bit wide ALU. Under program control, two locations, A and B, are simultaneously addressed from the 40 possible RAM locations. The two 25-bit words are fetched with the data from the A address passing through a binary shifter. This shifter allows scaling from 2^2 (a 2-bit left shift) to 2^{-13} (a 13-bit right shift). These values are then propagated to the ALU for processing with digital instructions specified by the program. The 25-bit result of that operation is loaded into the B address location of the RAM. When outputting a value, the nine most significant bits of a RAM location are loaded into the DAR. The DAR drives the D/A converter, whose output can be routed to any of eight analog outputs by the output demultiplexer and S&Hs.

What makes the 2920 fast enough for real-time processing is that the

analog operation, dual memory fetch, binary shift, ALU execution, and write back to RAM all take place in as little as 400 nsecs (depending on the clock rate of up to 10 MHz).

A fresh approach to analog circuit design

Despite its revolutionary implications, the 2920 does not replace any of the traditional functions of the analog designer. Instead, the 2920 and its development package increase design efficiency...while giving new options for design implementation.

Programming the 2920 to implement a particular system is done by first developing a system block diagram. Each block is then realized with assembly language routines, then assembled and simulated with the 2920 development support package. Instead of complex interfaces between subsystems, the 2920 lets you easily form blocks into the total system through software.

The end result is that manufacturers can offer customers a wider range of products with reduced cost, size and weight. And that designers can implement these new designs far more quickly than previous approaches.

Introducing the first general purpose analog processor

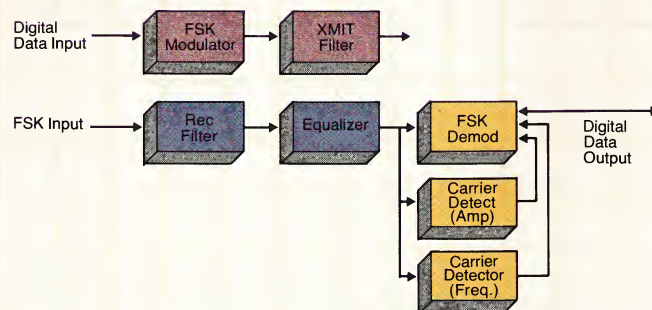
Designers can implement virtually any analog system in the DC to 10-kHz range with the programmable 2920 Signal Processor and its complete hardware and software development package. Applications are as broad as your imagination.

Use the 2920 as an economical alternative to developing custom analog circuits without the risks and commitments associated with special purpose components. It means unprecedented design flexibility, allowing you to make modifications, design improvements and add extra features by simply changing the 2920 program.

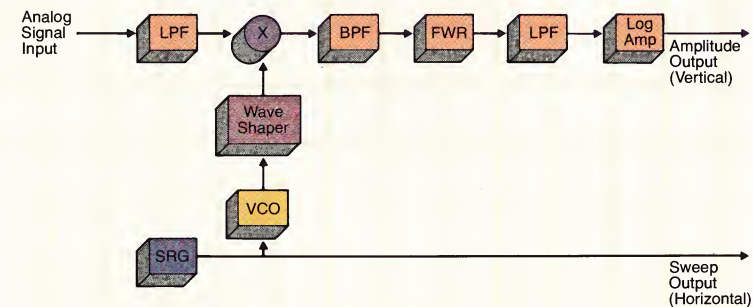
For example, the 2920 is an ideal single chip system solution for applications like modems and scanning spectrum analyzers. Or to implement DTMF receivers, phase lock loops, complex filters, linearizers, waveform generators, guidance and control modules, test and instrumentation circuits, speech processors, seismic processors, and medical instrumentation. Here are a few sample configurations:

Building a Single Chip Modem

The 2920's 192 instructions are enough to implement a 1200 bit-per-second frequency shift keyed (FSK) modem — complete with an FSK modulator, receive filters, selectable equalizer, FSK demodulator, amplitude and frequency carrier detection and strap options on a single IC (see block diagram). Higher speed modems using phase shift keyed (PSK) signaling can also use the 2920 to implement PSK modulators/demodulators and filters. You can even build adaptive equalizers by cascading 2920s with about 10 coefficients per device.



Single Chip Modem



Audio Scanning Spectrum Analyzer

Audio Scanning Spectrum Analyzer

A single 2920 is sufficient to implement an audio scanning spectrum analyzer (see block diagram). This analyzer is complete with an input filter, sweeping local oscillator, mixer, bandpass filter, envelope detector, and a precision logarithmic output amplifier. The 2920 implementation makes a straightforward design task of high stability and very high frequency resolution spectral analysis...yet uses only 75% of the EPROM storage.

Multi-Frequency Receiver/Analyzer

An example of the typical filter circuit possible with Intel's 2920 has the input signal bandlimited, amplitude normalized and then analyzed by a bank of bandpass filters (see block diagram). Filter outputs can be envelope detected and then buffered or compared to a threshold. In this example, the eight outputs are selected to be TTL compatible,

providing a digital interface. Analog outputs can also be made available using the output mode control pins.

Digital Filters made easy

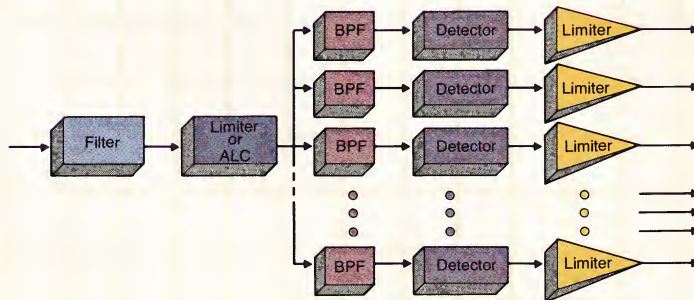
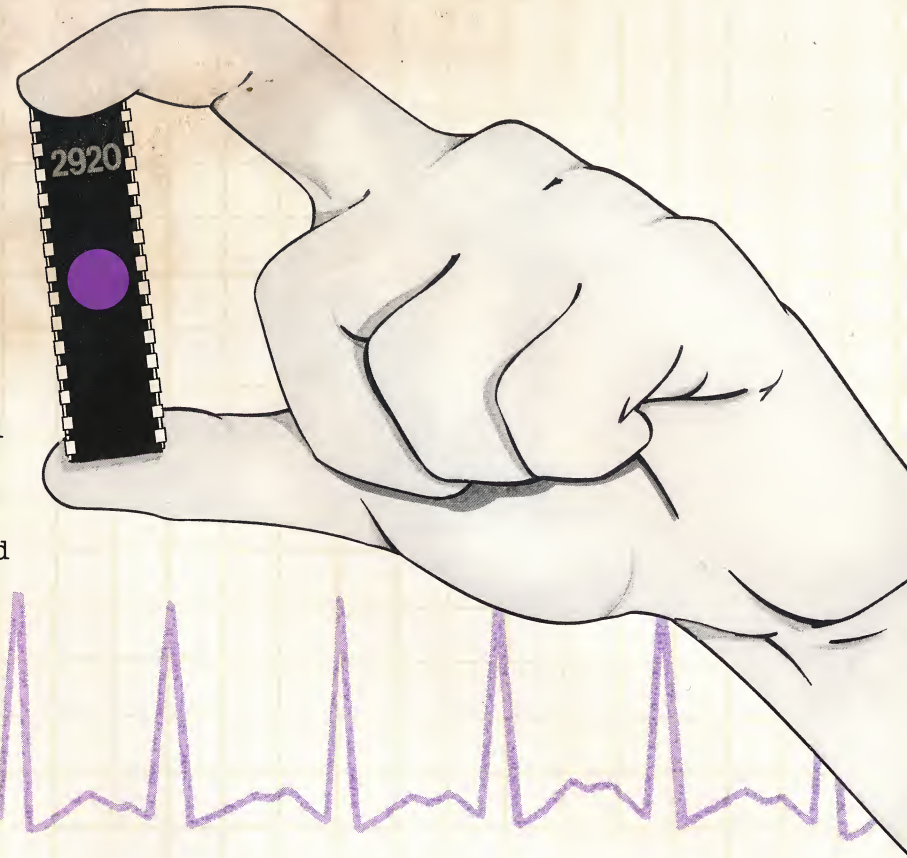
Use the 2920 to implement digital filters in almost any configuration and with very high complexity. The block diagram shows one example: a three-pole, two-zero, lowpass filter.

Several filters can be implemented, using up to a maximum of 40 delay elements with coefficients that require up to 192 instructions for implementation. A typical filter uses approximately 8-10 instructions per quadratic section (a complex pole or zero pair).

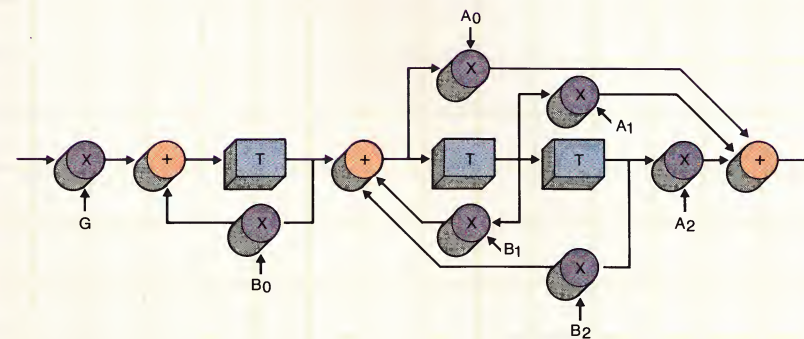
Typical uses for the 2920 in this kind of application include lowpass filters, bandpass filters, phase equalizers, and arbitrary transfer functions.

Digital Input/Output

Since the 2920's A/D conversion is under program control, a single bit of conversion can be implemented to read a logical level at one of the four analog inputs. Here, the converted amplitude bit takes on the same logical value as the input, which could then be used as data or as a control input by the Arithmetic Logic Unit (ALU). Serial digital bit streams can also be outputted by use of mode control pins, which convert the output pins to TTL compatible. Simple shift instructions and logical operations internal to the 2920 are used to perform the serial data movement at both input and output. The serial I/O of a 9-bit digital word requires approximately 25 instructions.



Multi-Frequency Receiver/Analyzer



Digital Filter



Intel delivers the development support to speed system implementation.

For simplicity in developing the 2920 system, Intel supplies an advanced design package of both hardware and software. Using these design tools, the typical design sequence for the 2920 is straightforward.

Our Intellec® Microcomputer Development System with SP20 Support Package furnishes the designer with a 2920 assembler, software simulator and EPROM programmer.

The diskette-based Intellec system is the key to the 2920's improved design efficiency. With this system, the designer can enter, review and edit with Intellec's keyboard and CRT. Intellec's accompanying operating software, called ISIS, provides the system utilities necessary for program development. The 2920's assembler and simulator are also diskette based, and run under ISIS' operating system software.

Typical development sequence

In the first step, you develop a detailed block diagram similar to those used for standard continuous analog designs.

Each of the function blocks is converted into 2920 code and arranged in proper sequence. With the Intellec Microcomputer Development System, each block can be assembled and simulated, either individually or as a system... depending on their complexity.

Intel's 2920 assembler translates user-written symbolic assembly language programs into machine code. This assembler generates several outputs, including comments, error/warning diagnostics, the number of RAM and ROM locations used, and a table of user-defined symbols. Ultimately, the machine code is used as an input to the software simulator, and to the EPROM programmer.

In the second step, you use the simulator to test the actual operation of the new program. Operating entirely in software, the 2920 simulator allows you to test and debug 2920 programs without building a breadboard. Initial

testing is accomplished by specifying input signals and simulating program execution.

Test, change, trace...effortlessly

If a problem occurs, the simulator's debug tools can be called into action, ready to test variables at different points in the circuit. Just use the simulator to stop or trace action at any point, and display registers and memory locations inside the 2920. Implement program changes, when necessary, with the simulator by directly changing the contents of the 2920 through the Intellec system keyboard. Then retest the revised program. That's how simple it is to test, change, trace or manipulate virtually every parameter of the 2920 as part of the debug procedure. You can document each step by using a line printer to generate hard copy. Or simply store your program on disk for later analysis.

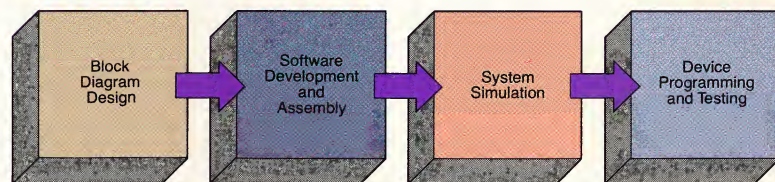
Once your system is complete in software, you can quickly transfer the program to the 2920 Signal Processor via our Intellec system and Universal PROM Programmer. With

your simulated program operating to specifications, you can load it into the 2920 chip for real-time testing.

How the 2920 total system approach gives you the competitive edge

With the 2920, it becomes evident how the advantage of digital processing in an analog environment can give you the edge. Since each production system will identically duplicate the original 2920 prototype without tweaking or tuning, you can convert the prototype into a production product. Forget drifting, aging, temperature effects or other typical problems that plague analog system production and reliability. You develop a digital code, so your system will be precisely duplicated in manufacturing.

All told, you've got the edge on a shorter development cycle. The edge on a new technology and vast new product opportunities. The edge it takes to speed your systems to market well ahead of the competition. The edge with a breakthrough in analog design... Intel's 2920 Signal Processor, Intel's SP20 Support Package, and the Intellec® Microcomputer Development System. They're all here today.



The 2920 Development Sequence

Intel's new 2920 product support

Everything you need to begin designing a new generation of real-time analog processing systems is available now.



The 2920 Family of single-chip analog signal processors offers a variety of speed versions for different applications.



The SP20 Support Package, including assembler, simulator, and the EPROM Programming Personality Card.



Intel's Intellec® Development System provides a keyboard, CRT, diskette based operating system, 64K RAM, and Universal PROM Programmer.



2920 Signal Processing Workshops

A typical workshop will cover such topics as:

- Sampled Data Systems and Digital Signal Processing
- 2920 Signal Processor Architecture
- Use of 2920 Assembly Language
- Operation of Intellec® Development Systems
- Operation of 2920 Assembler and Simulator
- Applications Design Labs

For more information, contact your local distributor or Intel sales office. Or write Intel Corporation, Literature Department, 3065 Bowers Avenue, Santa Clara, California 95051. Or call (408) 987-8080.



2920-10 SIGNAL PROCESSOR

- Real Time Digital Processing of Analog Signals
 - Nominal Signal Bandwidths from DC to 10KHz
 - Digital Processing Accuracy and Stability
 - Special Purpose Instruction Set for Signal Processing
 - INTELLEC® Compatible Development System Software and Hardware
 - Multiple Analog Inputs and Outputs
 - On-Chip Sample and Hold Circuits and D/A Converter
 - On-Chip EPROM: User Programmable and UV Erasable
 - On-Chip Scratch Pad Memory
 - Analog and/or TTL Output Waveforms, User Selectable
 - $\pm 5V$ Power Supplies
 - N-MOS Process
- 2920-10 ($T_{CYC} = 400$ nsec), 2920-16 ($T_{CYC} = 600$ nsec), 2920-18 ($T_{CYC} = 800$ nsec)

The Intel® 2920 Signal Processor is a programmable, single chip analog and digital signal processor specifically designed to replace analog subsystems in real time processing applications. Its instruction set plus the high precision (25 bits) digital arithmetic logic unit provides the capability to implement very complex subsystems. Typical functions performed by the 2920 include: Lowpass and Bandpass filters with up to 20 complex pole and/or zero pairs; Threshold Detectors; Limiters; Rectifiers; up to 25-bit multiplication and division; approximations to nonlinear functions such as square law and logarithm; logical operations; Input and Output multiplexing of signals; logical outputs for decision type processing; and analog outputs for multifrequency oscillators, waveform generators, etc. In addition, several 2920's may be cascaded for very complex processing applications with no loss in throughput rate.

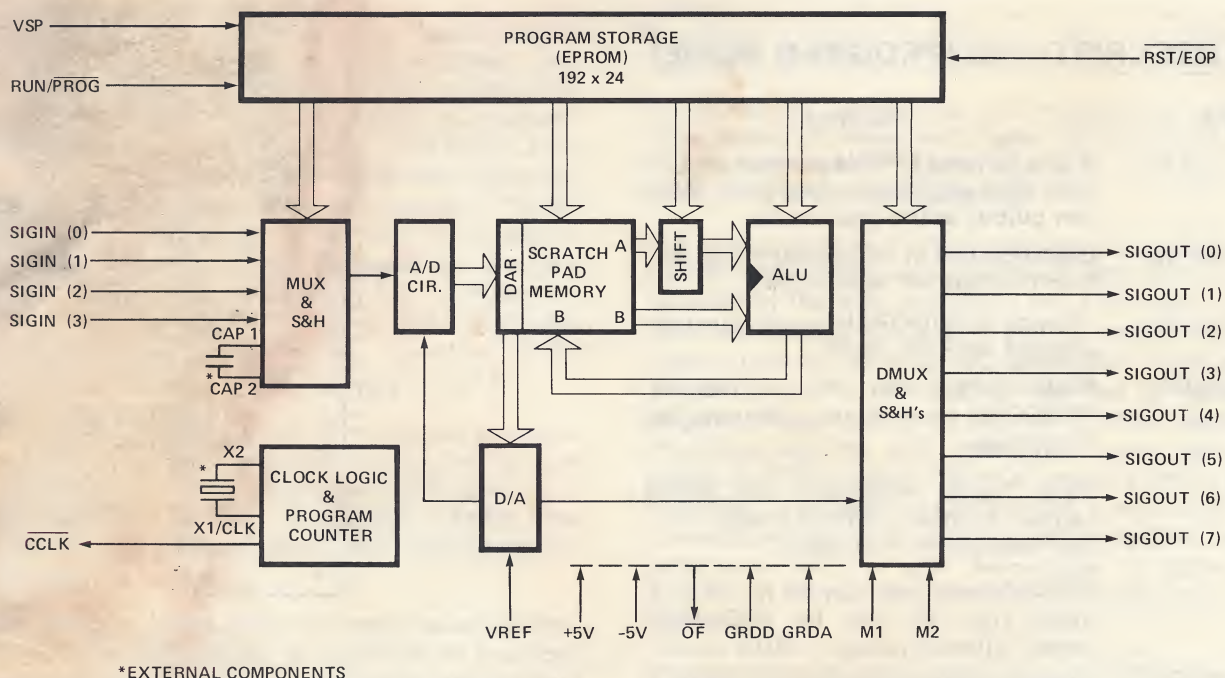


Figure 1. Functional Block Diagram (Run Mode).

PIN DESCRIPTIONS (RUN MODE)

Symbol	Function
SIGOUT	8 pins corresponding to the 8 demultiplexed analog outputs (0-7).
GRDA	Analog signal ground held at or near GRDD typically.
CAP ₁ & CAP ₂	External capacitor connections for the input signal sample and hold circuit
VREF	Input Reference Voltage.
SIGIN	4 pins corresponding to the 4 multiplexed analog inputs (0-3).
V _{BB}	Most negative power pin set at -5 volts during run mode (different voltage in program mode).
X1/CLK	Clock input when using external clock signals, oscillator input for external crystal when using internal clock.
X ₂	Oscillator input for external crystal when using internal clock.
GRDD	Digital ground.
V _{CC}	5 volts in run mode.
CCLK	Internal fetch cycle clock output. The falling edge designates the START of a new PROM fetch cycle. CCLK is 1/16 of X1/CLK rate.
RUN/PROG	Mode control tied to GRDD in run mode (different voltage in program mode).
RST/EOP	Low RST input initializes program fetch counter to first location. As an output it signifies EOP instruction present (open drain, active low).

Symbol	Function
OF	Indicates an overflow in the current ALU operation (open drain, active low).
VSP	EPROM power Pin 0 volts for RUN mode. (Different voltage in program mode).
M1, M2	Two pins which specify the output mode of the SIGOUT pins (see Table 4).

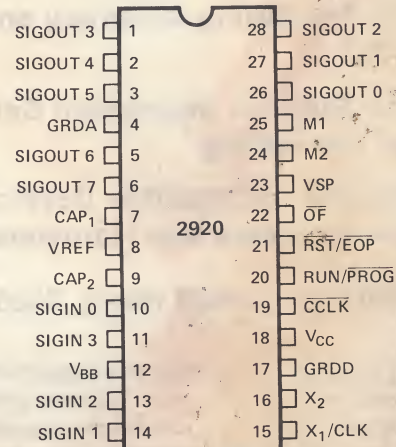


Figure 2. Run Mode Pin Configuration.

PIN DESCRIPTIONS (PROGRAM MODE)

Symbol	Function
D0,D1,D2,D3	4 pins carrying EPROM program data for both input and output (open drain, active low output; active high input).
V _{B1} , V _{B2} , V _{B3}	Digital ground in PROGRAM mode (different voltage for RUN mode).
V _{S1} , V _{S2} , V _{S3}	+5 volts in PROGRAM mode (function changes for RUN mode).
RUN/PROG	Mode control pin tied to V _{BB} for PROGRAM mode (voltage changes for RUN mode).
INCR	Input pulse increments the nibble (4-bits) counter in PROG mode (function changes in RUN mode).
VSP	EPROM power pin +5 volts for VERIFY mode and +25 volts for PROGRAM mode (different voltage in RUN mode).
PROG/VER	Controls EPROM bi-directional data bus for verify (low) or program (high).
RST	Input pulse resets nibble counter to position zero for start of programming.

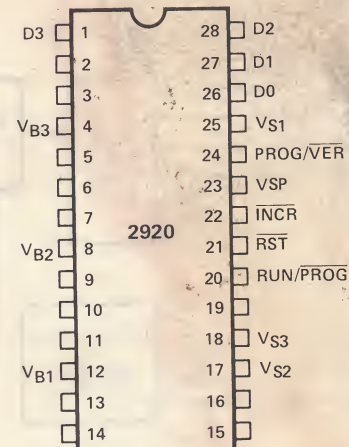


Figure 3. Program Mode Pin Configuration.

FUNCTIONAL DESCRIPTION

The Intel® 2920 is a programmable, single chip analog and digital signal processor which has been designed specifically to replace analog systems in real-time processing applications. The 2920 operates its analog circuitry simultaneously with the digital circuitry thereby achieving the efficiency and speed needed for real-time operation. Digital circuitry includes: EPROM program storage, RAM scratch pad memory, clock and timing circuitry, binary scaler, and the arithmetic logic unit (ALU). The analog circuitry is composed of 4 analog inputs, an input multiplexer, an input sample and hold (S&H), A/D and D/A converters, an output multiplexer, 8 analog outputs, and buffered output S&H's.

Once the EPROM is programmed, the 2920 is ready for operation as an analog subsystem. The following signal flow and operations can be described with reference to the functional block diagram shown in Figure 1.

Clock and Timing Logic — The 2920 can use an external clock or can generate its own clock with an external crystal placed across Pins 15 and 16. The program counter is incremented one instruction count for every 4 master clock cycles and continues to increment until it reaches a count of 191. Instructions are executed sequentially and no program jumps are provided. The sample rate is determined by the number of instructions in the program and the instruction cycle time. A 10 MHz clock (400 nsec instruction cycle time) and a full 192 instructions will result in a sample rate of 13,020 Hz.

Program Storage and Control — The EPROM is made up of 192 words with 24-bits per word. Each 24-bit word contains 6 instruction fields (see Table 1) which control the individual subsystems in the 2920.

RAM — The memory consists of a random-access read/write array organized as 40 words of 25-bits each. The address space is extended to provide constants and access to a register (DAR) for interfacing the memory-ALU with the analog conversion section. The RAM is a two port memory where the "A" location is Read Only and passes via a scaler to the ALU as one operand. The "B" location data passes to the ALU input as its second operand and the ALU result is written back to it. Both the RAM and the ALU represent data in two's complement format. All operations are performed in two's complement arithmetic. Program operations are simplified by assuming the binary point to the right of the sign bit.

An extended address space is used to generate constants within the program. It is accessed through the "A" port only and may be addressed using the last 16 locations of the "A" address field (i.e., "A" address 11XXXX). The constant is determined by the 4 least significant "A" address bits.

These 4 bits are treated as the 4 most significant bits at the input to the binary shifter. A sequence of extended addresses with shift operations can generate any constant up to 25 bits long.

The DAR is 9 bits wide and can be accessed in several ways. As a memory location, the DAR occupies the 9 most

significant bit positions of the 25-bit word and can be accessed as "A" and/or "B" port. The DAR output is also tied directly to the D/A converter inputs and is used as a successive approximation register for A/D conversion under control of the analog function instruction fields. Each bit position of the DAR can also be selected and tested for conditional arithmetic operations.

Binary Shifter — The 2920 has a binary shifter between the memory "A" port output and the ALU "A" operand input. This feature allows the "A" operand to be scaled by any magnitude between 2^2 and 2^{-13} (left shift 2 to right shift 13). When a number is shifted right vacated bit positions are filled with the sign bit. (2's complement arithmetic shift). Shift op codes are shown in Table 2.

ALU — The Arithmetic-Logic Unit calculates a 25-bit result based on an operation performed on the scaled "A" and the "B" operands delivered from memory. The 25-bit result is written back into the "B" memory location near the end of the instruction cycle. The ALU has logic to accommodate the left shift scaling. For arithmetic operations, this logic is used to calculate a 25-bit result for normal operations and to maintain the sign bit when an overflow occurs. An overflow occurs only when the magnitude of the result is larger than the largest number that can be stored in memory (25 bits). In that event, the result is set to the largest magnitude value with the correct sign. This overflow algorithm protects the continuity of the digitized analog signals and helps maintain the stability of the signal processing functions implemented. It is analogous to an overdriven amplifier going into saturation.

Instruction Set — The 2920 assembler uses the following program format to specify the 24-bit instruction word stored in the EPROM:

ALU INSTRUCTION (3 BITS)	B ADDRESS (6 BITS)	A ADDRESS (6 BITS)	SHIFT CODE (4 BITS)	ANALOG INSTRUCTION (5 BITS)
--------------------------------	--------------------------	--------------------------	---------------------------	-----------------------------------

All processing subsystems are implemented using a combination of analog and digital instructions to input and output signals and/or data, and to realize the processing functions respectively.

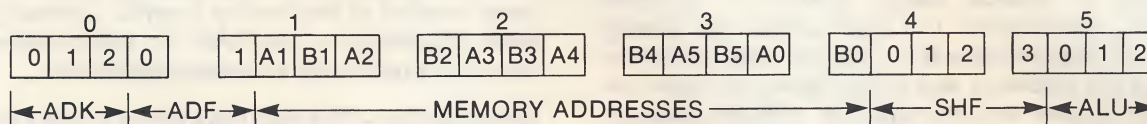
The analog input and output instructions are IN(K) and OUT(K) respectively. A sequence of IN(K) instructions followed by the sign conversion and amplitude conversion instructions CVTS and CVT(K) respectively are used to perform the input A/D conversion. A simple sequence of OUT(K) instructions is all that is needed to output a 9-bit amplitude on channel K. Other analog instructions are the EOP instruction which must be placed at ROM location 188, NOP which is simply a no-operation, and CNDS or CND(K) which are conditional operators which select and test a bit in the DAR for the conditional ADD or LDA instructions or define the destination of the carry bit for the conditional SUB instruction.

The ALU arithmetic instructions are ADD, SUB, LDA which perform the operations of addition, subtraction, and data transfer respectively. When these instructions are conditioned, they may be used to perform multiplication or division by a variable or data dependent switching.

Other digital instructions include the absolute value ABS, the absolute value and add ABA, and the ideal limit instruction LIM.

These instructions and their corresponding op codes are detailed in Table 3.

TABLE 1. NIBBLE ORGANIZATION FOR LOADING PROGRAM



Note: The input pins for each nibble bit from left to right are D0, D1, D2, D3.

TABLE 2. SHIFT OP CODES

Operation	Mnemonic	Op Code				Scale Factor
		3	2	1	0	
Shift Right 13 Bits	R13	1	1	0	0	2^{-13}
Shift Right 12 Bits	R12	1	0	1	1	2^{-12}
⋮	⋮					⋮
Shift Right 1 Bit	R01	0	0	0	0	2^{-1}
No Shift	R00	1	1	1	1	1
Shift Left 1 Bit	L01	1	1	1	0	2
Shift Left 2 Bits	L02	1	1	0	1	4

TABLE 3. INSTRUCTION SET AND OP CODES

Mnemonics		Op-Codes ^[1]			Operations	Notes
Code	Condition	ALU	ADF	ADK		
Digital Instructions		0,1,2	0,1	0,1,2		
ADD		011	↑	↑	$(Ax2^N) + B \rightarrow B$	
SUB		101	↑	↑	$B - (Ax2^N) \rightarrow B$	
LDA		111	↑	↑	$(Ax2^N) + 0 \rightarrow B$	
XOR		000	↑	↑	$(Ax2^N) \oplus B \rightarrow B$	
AND		100	↑	↑	$(Ax2^N) \cdot B \rightarrow B$	
ABS		110	↑	↑	$ (Ax2^N) \rightarrow B$	
ABA		001	[3]	[3]	$ (Ax2^N) + B \rightarrow B$	
LIM		010			$\text{Sign}(A) \rightarrow \pm \text{F.S.} \rightarrow B^{[4]}$	
ADD	CND() ^[2]	011	↑	↑	$(Ax2^N) + B \rightarrow B$	IFF DAR(K) = 1
					$B \rightarrow B$	IFF DAR(K) = 0
SUB	CND() ^{[2][8]}	101	↑	↑	$B - (Ax2^N) \rightarrow B$	& CY $\rightarrow$ DAR(K) IFF $CY_P = 1$
					$B + (Ax2^N) \rightarrow B$	& CY $\rightarrow$ DAR(K) IFF $CY_P = 0$
LDA	CND() ^[2]	111	↑	↑	$(Ax2^N) \rightarrow B$	IFF DAR(K) = 1
					$B \rightarrow B$	IFF DAR(K) = 0
ABA	CND() ^[9]	001			$(Ax2^N) + B \rightarrow B$	
XOR	CND() ^[9]	000			$(Ax2^N) \oplus B \rightarrow B$	
Analog Instructions						
IN(K)		↑	00	0-3	Signal Sample from Input Channel K	
OUT(K)		↑	10	0-7	D/A to Output Channel K	
CVTS		↑	00	6	Determine Sign Bit	
CVT(K)		↑	01	0-7	Perform A/D on Bit K	
EOP		[7]	00	5	Program Counter to Zero	
NOP		↑	00	4	No Operation	
CND(K)		↑	11	0-7	Select Bit K for Conditional Instructions	
CNDS		↑	00	7	Select Sign Bit for Conditional Instructions	

Notes: 1. Op codes ALU and ADF are in binary notation, ADK is in decimal notation and represents the value "K" when appropriate.

2. CND() can be either CND(K) or CNDS testing amplitude bits or the sign bit of the DAR respectively.

3. Determined by analog instructions below.

4. B is set to full scale (F.S.) amplitude with the same sign as the "A" port operand.

5. The previous carry bit (CY_P) is tested to determine the operation. The present carry bit (CY) is loaded into the Kth bit location of the DAR.

"Present carry (CY) is generated independent of overflow. It will represent the carry (CY) of a calculated 28-bit result."

6. EOP will also enable overflow correction if it was disabled during a program pass. The EOP must occur in ROM location 188.

7. Determined by digital instructions above.

8. For SUB CNDS operation $\bar{CY} \rightarrow \text{DAR}(S)$.

9. Does not affect DAR. In this case, CND is used with XOR /ABA to enable/disable the ALU overflow saturation algorithm. Use of either instruction causes the ALU output to roll over rather than go to full scale with sign bit preserved. An EOP instruction will also enable the ALU overflow saturation algorithm.

10. Clarification of CY_{OUT} sense for certain operations. For LDA, XOR, AND, ABS; $CY_{OUT} \rightarrow 0$.

Input Multiplexer and S&H — The input channels consist of four analog sampling switches which use a common external sampling capacitor. The external capacitor should be approximately 1000pF to yield an offset of less than $-1/2$ LSB and an acquisition time of less than 2400 nsec. The acquisition time is achieved in the program by using a sequence of IN(K) instructions equivalent to 2400 nsec (6 IN(K)'s for a 400 nsec instruction cycle).

A/D-D/A Converter — The successive approximation analog to digital conversion is performed under program control. It uses the CVTS instruction first to set the sign bit and the CVT(K) instruction to determine the value of the Kth DAR bit starting with the MSB. Two NOP's must follow each CVT(K) instruction. All A/D conversions of 2 bits or more must have "ADD DAR, KM2, RO, CND6" immediately following the "CVTS" instruction followed by 2 analog NOP's.

Output Demultiplexer and S&H's — The 2920's eight analog output channels include a sample and hold circuit per channel demultiplexed from a common, buffered DAC output. Two rules for outputting samples are: (1) No outputting should be done while writing to the DAR. (2) A sequence of NOP instructions equal to 2400 nsec should be used to settle the D/A converter and then a sequence of OUT(K) instruction equal to 2400 nsec will provide a settled S&H output on the Kth channel.

Conditional operations should *not* immediately precede or follow an OUT instruction. Otherwise a CND(K) may affect the value of the Kth output.

TTL Output — The SIGOUT(K) pins can be selected to be either analog out or TTL compatible as seen below. The

analog mode allows the full 9-bit D/A output to be present. The TTL mode requires an internal threshold be set under program control to yield a "0" or "1" decision. This output can be presented to the SIGOUT(K) pins and is compatible to a single TTL gate or equivalent. The internal threshold required is +1.6 volts for a high level output. An external pullup resistor to V_{CC} is also required.

TABLE 4. OUTPUT MODE FOR SIGOUT PINS AS FUNCTION OF M1 AND M2

M1	M2	SIGOUT Pins
5V	5V	0-7 Analog
5V	-5V	0-3 TTL, 4-7 Analog
-5V	5V	0-3 Analog, 4-7 TTL
-5V	-5V	0-7 TTL

Reference Voltage — The internal D/A converter requires a single positive reference voltage (V_{REF}) to establish its voltage range. This user supplied reference can range from 1V to 2V. The resulting input and output signal voltage range is $\pm V_{REF}$. For full 9-bit resolution and best linearity, $V_{REF} = 1V$. If the TTL output is required, $V_{REF} > 1.5V$ is necessary. The minimum voltage step (LSB) of the D/A converter is $V_{REF}/256$ volts. Voltage variations on V_{REF} will appear as noise to the D/A converter. It is therefore necessary to provide a noise free voltage source for the reference. The input signal voltage range is $(\pm V_{REF}) - 1/2$ LSB. The output signal voltage range is $(\pm V_{REF}) \times .93$ (min.).

SP20 DEVELOPMENT SUPPORT PACKAGE

This package includes the 2920 assembler, a software simulator, an EPROM programming personality card and a UPP adapter socket. Both 2920 assembler and software simulator run on the Intellec® Microcomputer Development System. The programming personality card provides the capability to program the 2920 EPROM on the Intellec peripheral—Universal Prom Programmer (UPP103).

The assembler translates symbolic 2920 assembly language programs into the machine operation codes. The codes can be loaded into the simulator for 2920 simulation or to the Universal PROM programmer for 2920 EPROM programming.

The simulator operates entirely in software, slower than the actual device, and allows the user to test and symbolically debug 2920 programs. The user can specify input signals, simulate program execution, set up breakpoints, display inputs and outputs, display and alter the contents of the 2920 registers and memory locations. The simulator can also stop or trace the program and give the user access to the key elements inside a 2920 for analyzing his program.

The assembler and simulator enable the designer to develop and test an entire program without putting it into a device. The 2920 designer works at the Intellec microcomputer development system rather than at a breadboard. The development system can store and recall programs or developed routines and aid in 2920 program design.

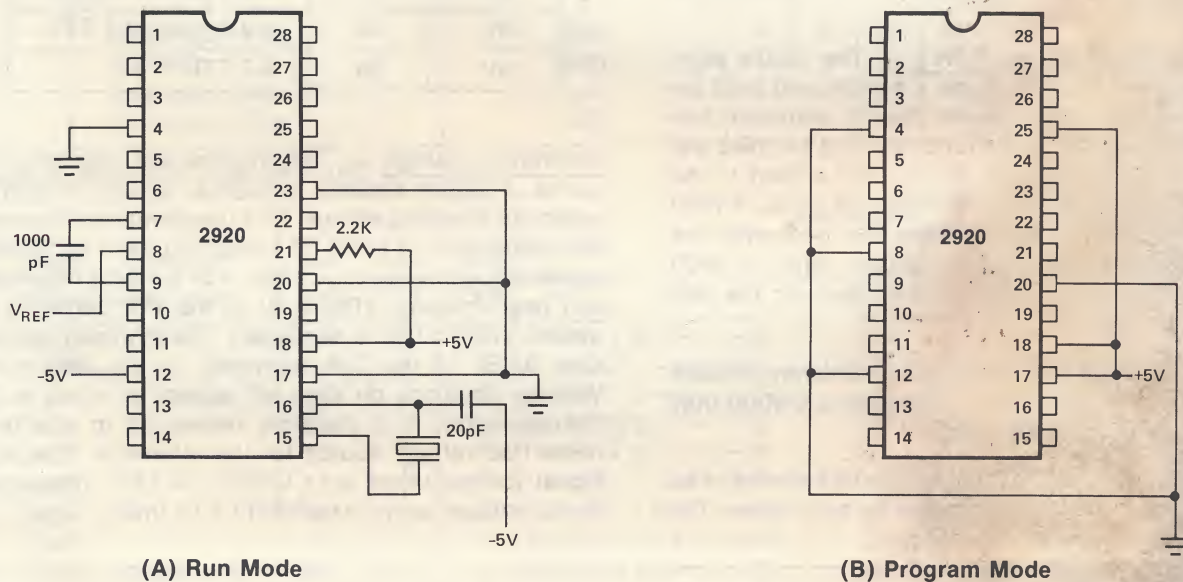
EPROM PROGRAMMING

The 2920 EPROM in the programming mode is arranged as a 1152 by 4-bit memory. Each instruction (24-bits) is loaded as 6 nibbles (4-bits) as seen in Table 3. Figure 6 shows the timing relationships of the signals required to program and verify the EPROM contents. In the program mode all voltages are referenced to V_{BB} which is set to digital ground, thereby allowing TTL logic to be used for controlling the programming cycle. The D pins are bi-directional with the direction controlled by the PROG/VER pin. A high level at the PROG/VER pin switches to input mode, a low to output mode (see Figure 6). This

feature allows the programmed data to be verified before going on to the next address.

The internal nibble counter is incremented during the falling edge of $\overline{INCR}$. 1152 $\overline{INCR}$ transitions will complete the full program cycle. To initialize at address (nibble) 0, $\overline{RST}$ must be pulsed low, then $\overline{INCR}$ can be issued. From then on, programming is accomplished according to Figure 6.

The RUN/ $\overline{PROG}$ pin must be tied to V_{BB} and VSP should be pulsed between +5V and +25 $\pm$ 1V at 15mA maximum. The D pins have an open drain in the output direction.



- Notes: 1. Analog grounding is connected to the GRDA lead. The GRDA and GRDD lead are not connected inside the 2920. An external connection is thus necessary outside the device to tie all the analog ground lines to the common return of the system GRDD. That external connection has to have a minimal impedance (ground plane) to avoid a DC offset in the 2920.
2. Pullup resistor at Pin 21 is only required if an EOP instruction is used.

Figure 4. Typical Power Hookups for Run and Program Mode

ABSOLUTE MAXIMUM RATINGS*

Temperature Under Bias	-40°C to +80°C
Storage Temperature	-65°C to +150°C
Supply Voltage with Respect to V_{BB}	-0.5V to +14V
All Input Voltages	-0.5V to (V_{CC} +1V)
Outputs	-1V to (V_{CC} +1V)
Power Dissipation	1.5W

*COMMENT

Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

D.C. AND OPERATING CHARACTERISTICS (Run Mode)
 $T_A = 0^\circ\text{C to } +70^\circ\text{C}$, $V_{CC} = 5\text{V}$, $V_{BB} = -5\text{V}$
ANALOG (SIGIN(K), SIGOUT(K))

Symbol	Parameter	Limits			Units	Test Conditions
		Min.	Typ.	Max.		
I_L	Input Leakage when Sampling			1	μA	$-2\text{V} < V_{IN} < +2\text{V}$
I_Z	Input Impedance when Sampling			1000	Ω	In Series with CAPN to GRDA
V_L	A-D Conversion Linearly			$\pm 1/2$	LSB	
V_{SH}	Sample to Hold Offset			$\pm 1/4$	LSB	
X_I	Input Crosstalk		TBD		dB	
V_{ID}	Input Droop Rate		TBD		$\text{V}/\mu\text{s}$	
T_A	Aperture		TBD		nsec	
R_I	Input Voltage Range	-VREF		+VREF	V	
Z_O	Output Impedance			800	Ω	
I_O	Output Drive Current		500		μA	20pF Load
D_O	Output Voltage Range	-VREF		+VREF	V	
V_{OD}	Output S/H Droop Rate		TBD		$\text{V}/\mu\text{s}$	
T_{OA}	Output Acquisition Time	2000	3200		ns	
X_O	Output Crosstalk		TBD		dB	
I_{VR}	Voltage Reference Current	120		230	μA	$V_{REF} = 2\text{V}$
V_{SR}	Power Supply Rejections		TBD		dB	
V_{REF}	Voltage Reference Levels	1		2	V	

DIGITAL ($\overline{OF}$, $\overline{CCLK}$, $\overline{EOP}$, SIGOUT(K) in TTL Mode)

I_{IL}	Low Level Input Current			10	μA	$V_{IN} < V_{IL}$
I_{IH}	High Level Input Current			10	μA	$V_{IN} > V_{IH}$
V_{IL}	Input Low Voltage			+0.8	V	
V_{IH}	Input High Voltage	2.0			V	
V_{IXL}	Input Low Voltage, XI/CLK	-3		-5	V	
V_{IXH}	Input High Voltage, XI/CLK	-1		0	V	
V_{OL}	Output Low Voltage			0.4	V	$I_{OL} = 2.5\text{mA}$

POWER DISSIPATION

I_{CC}	Operating Current			25	mA	$V_{CC} = 5\text{V} \pm 10\%$
I_{BB}	Operating Current			120	mA	$V_{BB} = 5\text{V} \pm 10\%$

A.C. CHARACTERISTICS (Run Mode)
 $T_A = 0^\circ\text{C to } +70^\circ\text{C}$, $V_{CC} = 5\text{V}$, $V_{BB} = -5\text{V}$

T_{CYC}	Instruction Cycle Period	2920-10	400		ns	4 clock cycles
		2920-16	600		ns	
		2920-18	800		ns	
T_{CE}	Cycle Start to EOP Valid	30			ns	
T_{EH}	$\overline{EOP}$ Hold Time	0.2		50	μs	$1/2 T_{CYC} \pm 25\%$ into 5.6k Ω , is pf Load
T_{FS}	$\overline{OF}$ Setup Time	30			ns	
T_{FH}	$\overline{OF}$ Release Time	20			ns	
T_{CW}	$\overline{CCLK}$ Pulse Width	400			ns	Equals T_{CYC}

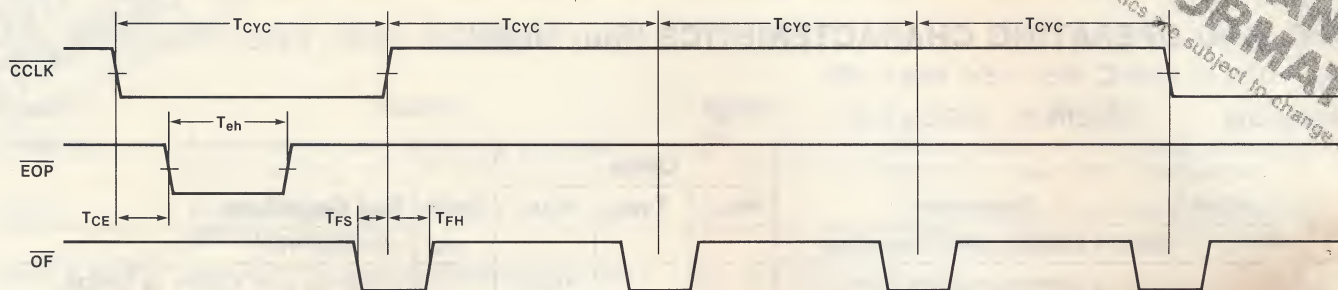


Figure 5. Run Mode Timing.

D.C. AND OPERATING CHARACTERISTICS (Program Mode) $T_A = 0^\circ\text{C to } +70^\circ\text{C}$, $V_{SS} = 5\text{V}$, $V_{BB} = 0\text{V}$

Symbol	Parameter	Limits			Unit	Test Conditions
		Min.	Typ.	Max.		
I_{TL}	Low Level Input Current			10	μA	$V_{IN} < V_{IL}$
I_{IH}	High Level Input Current			10	μA	$V_{IN} > V_{IH}$
V_{IL}	Input Low Voltage			0.8	V	
V_{IH}	Input High Voltage	2.0			V	
V_{OL}	Output Low Voltage			0.4	V	$I_{OL} = 2.5\text{mA}$
I_{SP}	Program Pulse Current			16	mA	Data Input = 0000
V_{P1}	Program Pulse ON Voltage	24	25	26	V	
V_{P2}	Program Pulse OFF Voltage		5		V	

POWER DISSIPATION

I_{SS}	Operating Current		TBD		mA	$V_{SS} = 5\text{V} + 10\%$
----------	-------------------	--	-----	--	----	-----------------------------

A.C. CHARACTERISTICS (Program Mode) $T_A = 0^\circ\text{C to } +70^\circ\text{C}$, $V_{SS} = 5\text{V}$, $V_{BB} = 0\text{V}$

T_{RW}	Reset Pulse Width	1000			ns	
T_{RS}	Reset to Increment Set-Up	200			ns	
T_{RH}	Reset Hold	300			ns	
T_{IW}	Increment Pulse Width	1			μs	
T_{RV}	Reset to Program Active	8			μs	
T_{VP}	Data In Set-Up to Prog. Pulse	2			μs	
T_{PW}	Program Pulse Width	50			ms	
$T_{PV(1)}$	Program to Verify Settling	1			μs	
T_{ACC}	Verify Access Time	20			μs	
T_{VI}	End of Verify to Increment	100			ns	

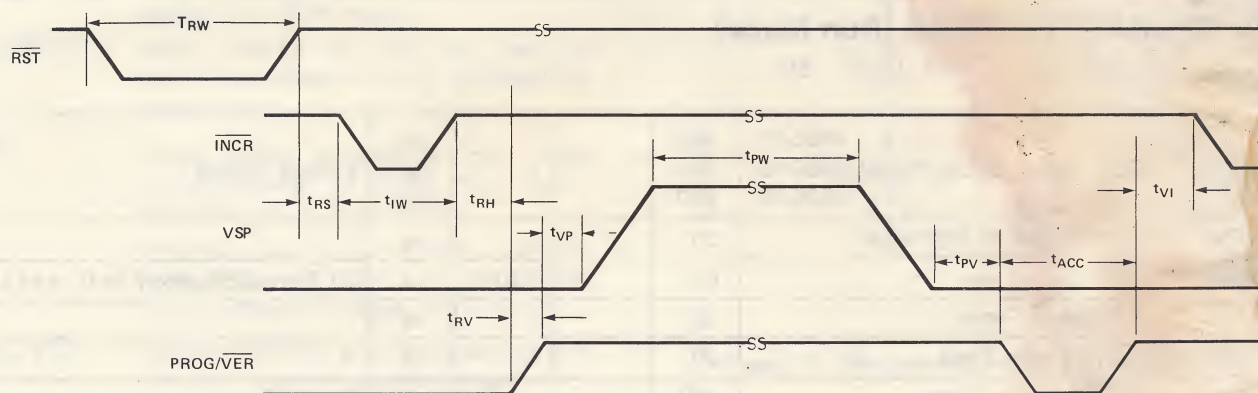
Note: 1. VSP must not undershoot 5V by more than 0.5V. Add undershoot settling time to T_{PV} .

Figure 6. Program Mode Timing.



JL 80

2920 SUPPORT PACKAGE

2920 Assembler for symbolic assembly language programming

2920 Simulator for system design and debug

Includes personality card for 2920 EPROM programming capability on the Intel Universal PROM Programmer

Simulator provides powerful debug capabilities, including conditional breakpoints, symbolic variable display and modification, and program trace

Extends Intellec® Microcomputer Development System to support 2920 development

Assembler output may be input to the Simulator or to the Intellec PROM Programmer for programming the 2920 EPROM

Takes advantage of powerful ISIS-II features including file handling and storage

The 2920 Support Package furnishes a 2920 assembler, 2920 software simulator and 2920 EPROM programming personality card. Both 2920 assembler and software simulator run on the Intellec® Microcomputer Development System. The programming personality card provides the capability to program the 2920 EPROM on the Intellec Universal PROM Programmer.

The assembler translates symbolic 2920 assembly language programs into the machine operation codes. The user can load the codes into the simulator for 2920 simulation or to the Universal PROM Programmer for 2920 EPROM programming.

The simulator, operating entirely in software, allows the user to test and symbolically debug 2920 programs. The user can specify input signals, simulate program execution, set up breakpoints, display input and output, and display and alter the contents of the 2920 registers and memory locations. The simulator can also stop or trace the program and constructively give the user access to the key elements inside a 2920 for analyzing his program.

The assembler and simulator enable the designer to develop and test an entire program without a complete prototype design. The 2920 designer works at the Intellec Microcomputer Development System rather than at a breadboard. The development system can store and recall programs or developed routines and aid in 2920 program design.



2920 Support Package

2920 ASSEMBLER

2920 program development on Intellec®
Microcomputer Development Systems

Produces Assembly Listing, Object Code
File, and Error Diagnostics

Translates symbolic assembly language
instructions into 2920 machine code

Output used for 2920 programming with
the Intellec PROM Programmer or the
2920 Simulator for program debug

The 2920 Assembler translates symbolic 2920 Assembly Language instructions into the appropriate machine operation codes. Through this facility, the programmer is able to symbolically program 2920 hardware operations. Compared to machine code, these symbolic references provide faster programming, easier debugging, and greater reliability.

The Assembler produces an object code file (executable machine code), a complete assembly listing, and error diagnostics. The object code output from the Assembler may be loaded directly into the Intel Universal PROM Programmer for programming the 2920 EPROM. The object code may also be loaded to the 2920 Simulator for 2920 system design and debug.

The 2920 Assembler runs under the ISIS-II Operating System on the Intellec Microcomputer Development Systems.

Sample 2920 Assembly Listing

ISIS-II 2920 ASSEMBLER X102

PAGE 1

ASSEMBLER INVOKED BY: AS2920 SAW.ASM DEBUG

SAWTOOTH WAVE GENERATOR

LINE LOC OBJECT SOURCE STATEMENT

```

1          $TITLE('SAWTOOTH WAVE GENERATOR')
2          ;
3          ;
4      0 0000EF      INO          ; SAMPLE INPUT CHANNEL 0
5      1 0000EF      INO
6      2 0000EF      INO
7      3 008AEB      SUB Y,KP1,INO ; SIMULTANEOUSLY CALCULATE SAWTOOTH
8      4 008A0A      SUB Y,KP1,R1,INO ; BY SUBTRACTING 3/16 FROM Y
9      5 0044EF      LDA DAR,Y,INO ; ALSO CHECK SIGN BIT OF Y
10     6 7A8AED      ADD Y,KP7,CNDS ; IF Y NEGATIVE START NEXT TOOTH
11     7 6000EF      CYTS         ; CONVERT SAMPLED INPUT TO DIGITAL (SIGN BIT)
12     8 7082EF      LDA Y,KP0,CNDS ; SUPPRESS SAWTOOTH IF INPUT WAS < 0
13     9 4044EF      LDA DAR,Y     ; PREPARE TO OUTPUT SAWTOOTH
14    10 4000EF      NOP          ; ANALOG LEVEL MUST SETTLE
15    11 4000EF      NOP
16    12 4000EF      NOP
17    13 8000EF      OUTO         ; OUTPUT SAWTOOTH
18    14 8000EF      OUTO
19    15 8000EF      OUTO
20    16 5000EF      EOP          ; PROGRAM WILL END IN THREE MORE INSTRUCTIONS
21    17 8000EF      OUTO
22    18 8000EF      OUTO
23    19 8000EF      OUTO
24          ;
25          END

```

SYMBOL:

VALUE:

Y

0

ASSEMBLY COMPLETE

ERRORS = 0

WARNINGS = 0

RAMSIZE = 1

ROMSIZE = 20

2920 SIMULATOR

Speeds test and debug of 2920 programs

Simulates 2920 internal operation

Operates on Intellec® Microcomputer Development Systems

Allows users to specify 2920 input signals, and display or alter ROM, RAM, and system variables

Output and internal data can be saved on disk for further analysis.

Provides ability to set breakpoints and to collect trace information

Easy-to-learn commands

Optionally uses the iSBC-310™ High-Speed Math Board for faster operation

The 2920 Simulator is a software facility that provides testing and symbolic debugging of 2920 programs in an Intellec Microcomputer Development Systems environment. The 2920 designers have the capability to specify the 2920 input signals, to set breakpoints, to collect and display 2920 input, output, system variables, and ROM and RAM data values during simulation. The 2920 Simulator accepts the hex format object files produced by the 2920 assembler. Output values and internal trace data may be saved on ISIS-II disk files for further analysis.

Functional Description

2920 Input Signal Specification

The four analog signal inputs to the 2920 processor can be specified as algebraic combinations of basic functions of time. The basic functions are SIN, COS, EXP, LOG, SQR, SAW, SQW, ABS.

2920 Simulation

The simulation of 2920 machine instructions is performed in software. All 2920 internal registers, memory, input values, output values, and other system variables can be examined and modified. The internal processing of the 2920 is simulated. Time constants for the sample and hold capacitors are assumed to be zero. Calculation of input signals is performed in single precision floating point. The speed of simulation varies with the complexity of the input signal, breakpoint setting, and trace condition. Exclusive of I/O time requirements, 2920 instructions will be simulated at a rate of approximately several hundred instructions per second.

Breakpoint Capabilities

After each instruction is simulated, the breakpoint is evaluated to determine whether to stop or continue simulation. Conditional breakpoints are also provided for debugging purposes. Simulation can be manually stopped at any time by pressing the ESC key on the Intellec console.

Trace Capabilities

Based on the qualifier's condition, trace data records can be collected during simulation. The trace data

records are stored in Intellec resident memory and are optionally written to the console for display or to a disk file for record.

Symbolic Debugging Capabilities

The 2920 Simulator allows the user to refer to program addresses symbolically. The user can load or save the symbols generated from the hex format object files or created during the debugging session. 2920 program memory in ROM can be disassembled, or filled with assembled instructions.

The 2920 Simulator is designed to provide users with powerful, easy-to-use commands. The user interfaces to the Simulator by entering commands to the Intellec console. The commands consist of one command line, terminated by one of the two line terminators — carriage return or line feed.

The 2920 Simulator offers two types of commands:

Simulation and Control Commands

Command	Operation
Simulate	Starts simulation of the input signals and the 2920 program in simulated ROM memory. Initial setting is "FOREVER."
Trace	Controls the trace selection. Initial setting is "TIME."
Qualifier	Sets qualifier condition during trace. Initial setting is "ALWAYS."
Breakpoint	Sets breakpoint condition during simulation. Initial setting is "NEVER."

2920 SUPPORT PACKAGE

Interrogation and Utility Commands

Command	Operation
Display	Displays the values of symbols, RAM, ROM, input, output, registers and system variables.
Change	Alters the values of symbols, RAM, ROM, input, register and system variables.
Base	Establishes the mode of display for output data.
Suffix	Establishes the mode of display for input data.
Load	Fetches user symbol table and object code from input device.
Save	Sends user symbol table and object code to output device.
Define	Enters symbol name and value to user symbol table.
Console	Controls the console on/off display.
List	Defines list device.
Exit	Returns program control to ISIS-II.
Evaluate	Converts expression to equivalent values in binary, decimal, and hex.
Remove	Deletes symbols from symbol table.
Help	Provides a brief summary of the syntax for the command languages.

Keyword References

The 2920 Simulator provides users with keyword references to gain access to all of the numeric valued system variables including simulated 2920's memory, register, status flags and input/output. These keyword references can function as the evaluation command, display command, and change command.

• 2920 Processor Keyword References

IN0	Analog input 0 in volts
IN1	Analog input 1 in volts
IN2	Analog input 2 in volts
IN3	Analog input 3 in volts
OUT0	Analog output 0 in volts (read only)
OUT1	Analog output 1 in volts (read only)
OUT2	Analog output 2 in volts (read only)
OUT3	Analog output 3 in volts (read only)
OUT4	Analog output 4 in volts (read only)
OUT5	Analog output 5 in volts (read only)
OUT6	Analog output 6 in volts (read only)
OUT7	Analog output 7 in volts (read only)
IN	Sampled and held analog input signal in volts
DAR	Digital to analog register (RAM location 40)
PC	Program counter (integer 1 to 192)
CY	Carry (integer 0 or 1)
OVF	Overflow (integer 0 or 1, read only)
OVE	Overflow enable (integer 0 or 1)

• Software Simulator Keyword References

TIME	Elapsed simulated time in seconds (read only)
TQUAL	Time when the qualifier last matched in seconds (read only)
COUNT	Number of instructions simulated since last SIMULATE command (integer, read only)
BUFFER SIZE	Number of trace data records (integer, read only)
TINST	Time between successive instructions in seconds (read only)
SIZE	Number of instructions in program disregarding actual EOP placement
TPROG	Time between successive program passes in seconds
VREF	Reference analog level voltage in volts

The above keyword references are designed to aid 2920 program debugging.

ISIS Compatibilities

The 2920 software simulator runs under the ISIS "submit" facility. The 2920 software simulator uses the ISIS-II line editing capabilities to correct errors in an input line on the Intellec console.

Sample 2920 Simulation Session

```

-SM2920
*LIST SAN.LOG      ; SAVE OUTPUT FROM THIS SESSION IN LOG FILE
*LOAD SAN.HEX      ; LOAD SAWTOOTH GENERATOR ASSEMBLED WITH AS2920
*FROM 0 TO 5       ; EXAMINE FIRST PART OF PROGRAM
ROM 000 = LDA .Y, Y, ROM, IN0
ROM 001 = LDA .Y, Y, ROM, IN0
ROM 002 = LDA .Y, Y, ROM, IN0
ROM 003 = SUB .Y, KP1, ROM, IN0
ROM 004 = SUB .Y, KP1, ROM, IN0
ROM 005 = LDA DAR, Y, ROM, IN0
*IN0 = SIN(4000*TIME) ; SPECIFY INPUT SIGNAL TO BE SIMULATED
*QUALIFIER = PC=0    ; COLLECT TRACE ONLY ONCE PER PROGRAM PASS
*TRACE = TIME, IN0, OUT0 ; TRACE THESE THREE ITEMS
*SIMULATE FROM 0 TILL TIME>TPI/4000 ; SIMULATE FOR A WHILE
TIME          IN0          OUT0
SIMULATION BEGUN
0.00009600    0.37463213    0.25000000
0.00019200    0.69469800    0.06250000
0.00028800    0.91357910    0.75000000
0.00038400    0.99939463    0.56250000
0.00048000    0.93964551    0.37500000
0.00057600    0.74303413    0.18750000
0.00067200    0.43819771    0.00000000
0.00076800    0.06953646    0.68750000
0.00086400    -0.30925317    0.00000000
0.00096000    -0.64299883    0.00000000
0.00105600    -0.88308994    0.00000000
0.00115200    -0.99455645    0.00000000
0.00124800    -0.96116289    0.00000000
0.00134400    -0.78777319    0.00000000
0.00144000    -0.49964165    0.00000000
0.00153600    -0.13873627    0.00000000
SIMULATION TERMINATED
*PROGRAM WORKS
*EXIT

```


SPECIFICATIONS

Operating Environment

Required Hardware

Intellec® Microcomputer Development System
— Model 800 or 888
— Series II Model 220 or 230
64K Bytes of RAM Memory
Universal PROM Programmer — UPP-103
One or two Floppy Disk Drives
— Single or Double Density
System Console
— CRT or interactive hardcopy device

Required Software

ISIS-II Diskette Operating System

Optional Hardware

Line Printer
ISBC-310 High-Speed Mathematics Unit

Optional Software

FORTRAN-80 (Product Code MDS-301*)

Documentation Package

Universal PROM Programmer User's Manual (9800819)
2920 Assembly Language Manual (9800987)
2920 Simulator Operating Instructions (9800988)

Shipping Media

Flexible Diskettes
— Single and Double Density
— Universal PROM Programming Personality Card

ORDERING INFORMATION

Product Code	Description
--------------	-------------

SP20-KIT	SP20 Support Package Includes 2920 software support package (MDS-329*) and 2920 EPROM programmer personality card (UPP-820)
----------	--

* MDS is an ordering code only and is not used as a product name or trademark.
MDS® is a registered trademark of Mohawk Data Sciences Corporation.



INTEL CORPORATION, 3065 Bowers Avenue, Santa Clara, CA 95051 (408) 987-8080

Printed in U.S.A./H173/1079/25K BL